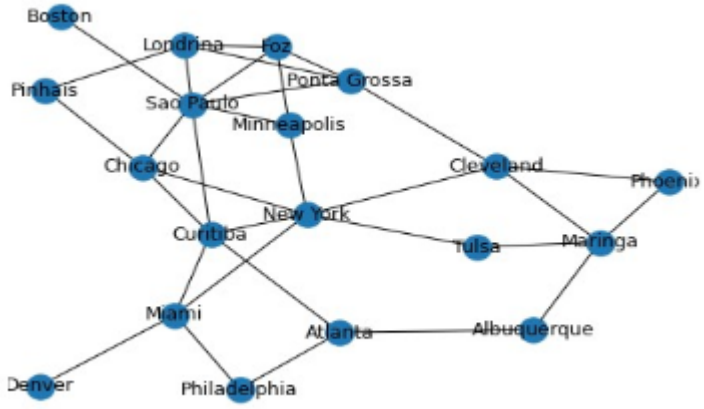
Jayme de Queiroz

1- A Dream Airlines tem o seguinte mapa de rotas para as cidades que atende, onde cada par de cidades tem serviço em ambas as direções entre as cidades:

Albuquerque - Atlanta
Chicago - New York
Chicago - Pinhais
Curitiba - Atlanta
Curitiba - Chicago
Curitiba - Miami
Curitiba - New York
Curitiba - Sao Paulo
Londrina - Foz
Maringa - Albuquerque
Maringa - Cleveland
Miami - Denver
Miami - New York
Miami - Philadelphia
Minneapolis - Foz
New York - Cleveland
New York - Minneapolis
Philadelphia - Atlanta
Phoenix - Cleveland
Phoenix - Maringa
Pinhais - Londrina
Ponta Grossa - Cleveland
Ponta Grossa - Foz
Ponta Grossa - Londrina
Sao Paulo - Boston
Sao Paulo - Chicago
Sao Paulo - Foz
Sao Paulo - Londrina
Sao Paulo - Minneapolis
Sao Paulo - Ponta Grossa
Tulsa - Maringa
Tulsa - New York

Construa um grafo apropriado que represente esses relacionamentos utilizando o Networkx. A rede produzida deve ficar mais ou menos assim:



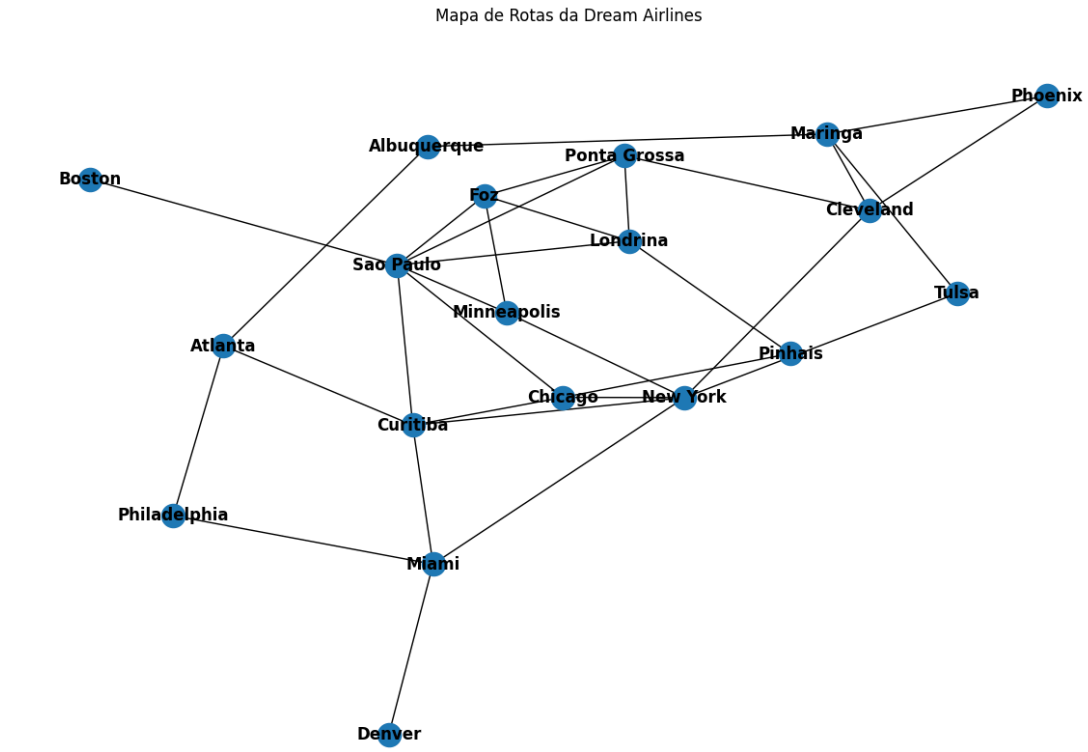
In [ ]:
```
import networkx as nx
import matplotlib.pyplot as plt
```

In [ ]:
```
# Criando um grafo direcionado
G = nx.Graph()

# Adicionando arestas com os pares de cidades
routes = [
    ("Albuquerque", "Atlanta"),
    ("Chicago", "New York"),
    ("Chicago", "Pinhais"),
    ("Curitiba", "Atlanta"),
    ("Curitiba", "Chicago"),
    ("Curitiba", "Miami"),
    ("Curitiba", "New York"),
    ("Curitiba", "Sao Paulo"),
    ("Londrina", "Foz"),
    ("Maringa", "Albuquerque"),
    ("Maringa", "Cleveland"),
    ("Miami", "Denver"),
    ("Miami", "New York"),
    ("Miami", "Philadelphia"),
    ("Minneapolis", "Foz"),
    ("New York", "Cleveland"),
    ("New York", "Minneapolis"),
    ("Philadelphia", "Atlanta"),
    ("Phoenix", "Cleveland"),
    ("Phoenix", "Maringa"),
    ("Pinhais", "Londrina"),
    ("Ponta Grossa", "Cleveland"),
    ("Ponta Grossa", "Foz"),
    ("Ponta Grossa", "Londrina"),
    ("Sao Paulo", "Boston"),
    ("Sao Paulo", "Chicago"),
    ("Sao Paulo", "Foz"),
    ("Sao Paulo", "Londrina"),
    ("Sao Paulo", "Minneapolis"),
    ("Sao Paulo", "Ponta Grossa"),
    ("Tulsa", "Maringa"),
    ("Tulsa", "New York")
]

# Adicionando arestas ao grafo
G.add_edges_from(routes)
```

In [ ]:
```
# Plotando o grafo
plt.figure(figsize=(12, 8))
nx.draw(G, with_labels=True, font_weight='bold', node_size=300)
plt.title("Mapa de Rotas da Dream Airlines")
plt.show()
```



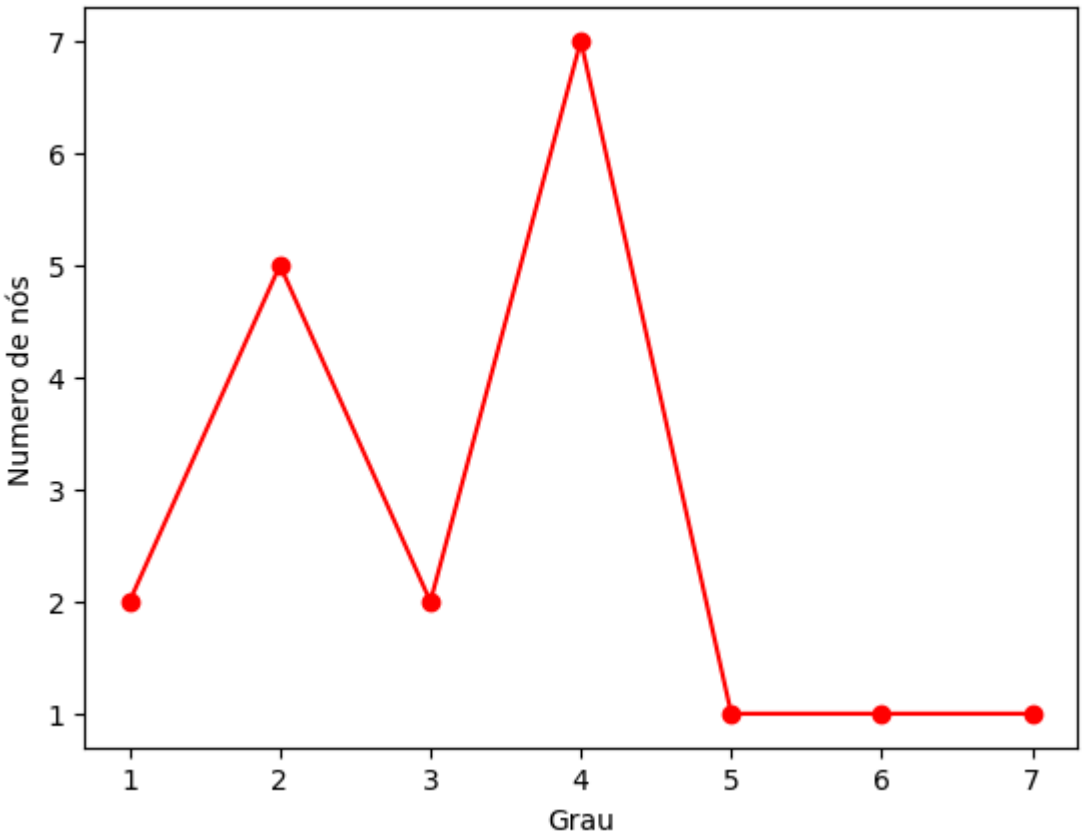
In [ ]:
```
dicDegree = dict(G.degree() )
valores = sorted(set(dicDegree.values()))
valores

hist = [list(dicDegree.values()).count(x) for x in valores]


plt.plot(valores, hist, 'ro-')

plt.xlabel('Grau')
plt.ylabel('Numero de nós')
```

Out[ ]:
```
Text(0, 0.5, 'Numero de nós')
```



2- Com base na rede construída no exercício 1, faça o que é proposto abaixo.\

a- Qual é o número máximo de saltos que um passageiro pode fazer em uma única viagem entre duas cidades atendidas?
b- Qual é o coeficiente de clusterização de Curitiba? E da rede geral?

In [ ]:
```
# a) Número máximo de saltos
max_hops = nx.diameter(G)
print("a) O número máximo de saltos em uma única viagem entre duas cidades atend

# b) Coeficiente de clusterização de Curitiba
#    Coeficiente de clusterização médio da rede geral
average_clustering = nx.average_clustering(G)
curitiba_clustering = nx.clustering(G, "Curitiba")
print(f"b) O coeficiente de clusterização de Curitiba é:{curitiba_clustering:.4f
```

```
a) O número máximo de saltos em uma única viagem entre duas cidades atendidas é:
4
b) O coeficiente de clusterização de Curitiba é:0.3000, e o da rede geral é 0.237
1
```

3- Estenda a rede criada na questão 1

a- Adicione o atributo nos nós chamado Country, onde o valor se refere ao país onde cada cidade está localizada.

In [ ]:
```
# Adicionando atributo "Country" aos nós
country_mapping = {
    "Albuquerque": "USA",
    "Atlanta": "USA",
```

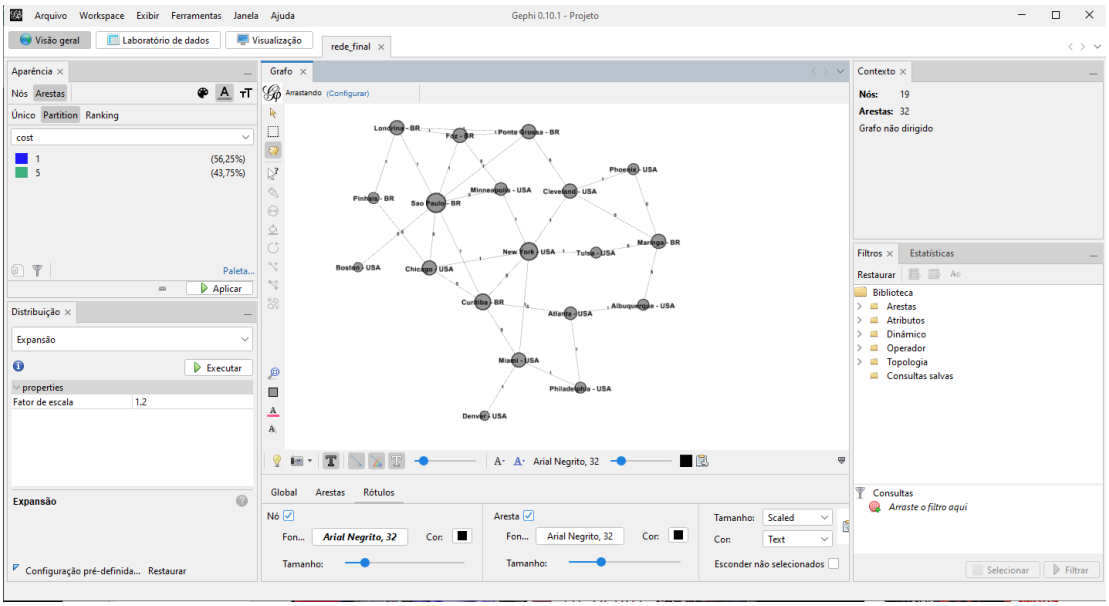
```
    "Chicago": "USA",
    "Pinhais": "BR",
    "Curitiba": "BR",
    "Miami": "USA",
    "New York": "USA",
    "Sao Paulo": "BR",
    "Londrina": "BR",
    "Foz": "BR",
    "Maringa": "BR",
    "Cleveland": "USA",
    "Denver": "USA",
    "Minneapolis": "USA",
    "Philadelphia": "USA",
    "Boston": "USA",
    "Phoenix": "USA",
    "Ponta Grossa": "BR",
    "Tulsa": "USA"
}

nx.set_node_attributes(G, country_mapping, "Country")
```

b- Adicione o atributo nas arestas chamado Cost. Se uma arestas representa um voo internacional esse valor deve ser 5, se ela representa um voo nacional o valor deve ser 1.

In [ ]:
```
# Adicionando atributo "Cost" às arestas
for edge in G.edges():
    source = edge[0]
    target = edge[1]
    if G.nodes[source]["Country"] == G.nodes[target]["Country"]:
        G.edges[source, target]["Cost"] = 1  # Voo nacional
    else:
        G.edges[source, target]["Cost"] = 5  # Voo internacional
```

4 – Exporte a rede final criada na questão 3 no formato .GML. Confira se todas as informações pedidas anteriormente estão disponíveis nesse arquivo.



In [ ]:
```
# Exportar a rede final para o formato .GML
nx.write_gml(G, "rede_final.gml")
```

```python
# Ler o arquivo .GML para verificar se as informações foram corretamente exporta
G_from_file = nx.read_gml("rede_final.gml")

# Verificar se as informações dos nós e arestas estão corretas
print("Atributos dos nós:")
print(f"G_from_file{G_from_file.nodes(data=True)}")
print(f"G          {G.nodes(data=True)}")
print("\nAtributos das arestas:")
print(f"G_from_file{G_from_file.edges(data=True)}")
print(f"           {G.edges(data=True)}")
```

```
Atributos dos nós:
G_from_file[('Albuquerque', {'Country': 'USA'}), ('Atlanta', {'Country': 'USA'}),
('Chicago', {'Country': 'USA'}), ('New York', {'Country': 'USA'}), ('Pinhais',
{'Country': 'BR'}), ('Curitiba', {'Country': 'BR'}), ('Miami', {'Country': 'US
A'}), ('Sao Paulo', {'Country': 'BR'}), ('Londrina', {'Country': 'BR'}), ('Foz',
{'Country': 'BR'}), ('Maringa', {'Country': 'BR'}), ('Cleveland', {'Country': 'US
A'}), ('Denver', {'Country': 'USA'}), ('Philadelphia', {'Country': 'USA'}), ('Min
neapolis', {'Country': 'USA'}), ('Phoenix', {'Country': 'USA'}), ('Ponta Grossa',
{'Country': 'BR'}), ('Boston', {'Country': 'USA'}), ('Tulsa', {'Country': 'US
A'})]
G          [('Albuquerque', {'Country': 'USA'}), ('Atlanta', {'Country': 'USA'}),
('Chicago', {'Country': 'USA'}), ('New York', {'Country': 'USA'}), ('Pinhais',
{'Country': 'BR'}), ('Curitiba', {'Country': 'BR'}), ('Miami', {'Country': 'US
A'}), ('Sao Paulo', {'Country': 'BR'}), ('Londrina', {'Country': 'BR'}), ('Foz',
{'Country': 'BR'}), ('Maringa', {'Country': 'BR'}), ('Cleveland', {'Country': 'US
A'}), ('Denver', {'Country': 'USA'}), ('Philadelphia', {'Country': 'USA'}), ('Min
neapolis', {'Country': 'USA'}), ('Phoenix', {'Country': 'USA'}), ('Ponta Grossa',
{'Country': 'BR'}), ('Boston', {'Country': 'USA'}), ('Tulsa', {'Country': 'US
A'})]

Atributos das arestas:
G_from_file[('Albuquerque', 'Atlanta', {'Cost': 1}), ('Albuquerque', 'Maringa',
{'Cost': 5}), ('Atlanta', 'Curitiba', {'Cost': 5}), ('Atlanta', 'Philadelphia',
{'Cost': 1}), ('Chicago', 'New York', {'Cost': 1}), ('Chicago', 'Pinhais', {'Cos
t': 5}), ('Chicago', 'Curitiba', {'Cost': 5}), ('Chicago', 'Sao Paulo', {'Cost':
5}), ('New York', 'Curitiba', {'Cost': 5}), ('New York', 'Miami', {'Cost': 1}),
('New York', 'Cleveland', {'Cost': 1}), ('New York', 'Minneapolis', {'Cost': 1}),
('New York', 'Tulsa', {'Cost': 1}), ('Pinhais', 'Londrina', {'Cost': 1}), ('Curit
iba', 'Miami', {'Cost': 5}), ('Curitiba', 'Sao Paulo', {'Cost': 1}), ('Miami', 'D
enver', {'Cost': 1}), ('Miami', 'Philadelphia', {'Cost': 1}), ('Sao Paulo', 'Bost
on', {'Cost': 5}), ('Sao Paulo', 'Foz', {'Cost': 1}), ('Sao Paulo', 'Londrina',
{'Cost': 1}), ('Sao Paulo', 'Minneapolis', {'Cost': 5}), ('Sao Paulo', 'Ponta Gro
ssa', {'Cost': 1}), ('Londrina', 'Foz', {'Cost': 1}), ('Londrina', 'Ponta Gross
a', {'Cost': 1}), ('Foz', 'Minneapolis', {'Cost': 5}), ('Foz', 'Ponta Grossa',
{'Cost': 1}), ('Maringa', 'Cleveland', {'Cost': 5}), ('Maringa', 'Phoenix', {'Cos
t': 5}), ('Maringa', 'Tulsa', {'Cost': 5}), ('Cleveland', 'Phoenix', {'Cost':
1}), ('Cleveland', 'Ponta Grossa', {'Cost': 5})]
          [('Albuquerque', 'Atlanta', {'Cost': 1}), ('Albuquerque', 'Maringa',
{'Cost': 5}), ('Atlanta', 'Curitiba', {'Cost': 5}), ('Atlanta', 'Philadelphia',
{'Cost': 1}), ('Chicago', 'New York', {'Cost': 1}), ('Chicago', 'Pinhais', {'Cos
t': 5}), ('Chicago', 'Curitiba', {'Cost': 5}), ('Chicago', 'Sao Paulo', {'Cost':
5}), ('New York', 'Curitiba', {'Cost': 5}), ('New York', 'Miami', {'Cost': 1}),
('New York', 'Cleveland', {'Cost': 1}), ('New York', 'Minneapolis', {'Cost': 1}),
('New York', 'Tulsa', {'Cost': 1}), ('Pinhais', 'Londrina', {'Cost': 1}), ('Curit
iba', 'Miami', {'Cost': 5}), ('Curitiba', 'Sao Paulo', {'Cost': 1}), ('Miami', 'D
enver', {'Cost': 1}), ('Miami', 'Philadelphia', {'Cost': 1}), ('Sao Paulo', 'Bost
on', {'Cost': 5}), ('Sao Paulo', 'Foz', {'Cost': 1}), ('Sao Paulo', 'Londrina',
{'Cost': 1}), ('Sao Paulo', 'Minneapolis', {'Cost': 5}), ('Sao Paulo', 'Ponta Gro
ssa', {'Cost': 1}), ('Londrina', 'Foz', {'Cost': 1}), ('Londrina', 'Ponta Gross
a', {'Cost': 1}), ('Foz', 'Minneapolis', {'Cost': 5}), ('Foz', 'Ponta Grossa',
{'Cost': 1}), ('Maringa', 'Cleveland', {'Cost': 5}), ('Maringa', 'Phoenix', {'Cos
t': 5}), ('Maringa', 'Tulsa', {'Cost': 5}), ('Cleveland', 'Phoenix', {'Cost':
1}), ('Cleveland', 'Ponta Grossa', {'Cost': 5})]
```

In [ ]:

```python
# Draw the graph with labels and attributes
plt.figure(figsize=(12, 8))

# Draw nodes with labels and attributes
pos = nx.spring_layout(G)  # Positioning nodes
nx.draw(G, pos, with_labels=False, node_size=1000)
```

```python
# Draw node attributes (Country) side by side with node labels
node_attributes = nx.get_node_attributes(G, 'Country')
for node, (x, y) in pos.items():
    plt.text(x, y, f"{node}\n{node_attributes[node]}", fontsize=10, ha='center',

# Draw edge attributes
edge_labels = nx.get_edge_attributes(G, 'Cost')
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels)

plt.title("Grafo com atributos")
plt.axis('off')  # Turn off axis
plt.show()
```